Learning Spike time codes through Morphological Learning with Binary Synapses

Subhrajit Roy, *Student Member, IEEE*, Phyo Phyo San, Shaista Hussain, Lee Wang Wei and Arindam Basu*, *Member, IEEE*

***Abstract*—In this paper, a neuron with nonlinear dendrites (NNLD) and binary synapses that is able to learn temporal features of spike input patterns is considered. Since binary synapses are considered, learning happens through formation and elimination of connections between the inputs and the dendritic branches to modify the structure or "morphology" of the NNLD. A morphological learning algorithm inspired by the 'Tempotron', i.e., a recently proposed temporal learning algorithm–is presented in this work. Unlike 'Tempotron', the proposed learning rule uses a technique to automatically adapt the NNLD threshold during training. Experimental results indicate that our NNLD with 1-bit synapses can obtain similar accuracy as a traditional Tempotron with 4-bit synapses in classifying single spike random latency and pair-wise synchrony patterns. Hence, the proposed method is better suited for robust hardware implementation in the presence of statistical variations. We also present results of applying this rule to real life spike classification problems from the field of tactile sensing.**

***Index Terms*—spiking neuron, tempotron, binary synapse, dendrites, plasticity, learning**

I. Introduction

Though the representation of stimulus by the neurons in our brain is a topic of much ongoing research and debate, it is widely believed that the timing of the action potentials or spikes fired by these neurons carry important information [1]. Spike latency codes i.e. delay in the spike time after stimulus presentation have been suggested for tactile, olfactory and retinal systems [2]. They are also thought to offer significant advantages in terms of reducing power needed for communicating spikes as well as allowing rapid processing of inputs. Hence, neuromorphic engineers, who aim to mimic the brain's processing capabilities in silicon, have also been interested in spike timing based neural networks.

Several analog CMOS integrated circuits operating in the sub-threshold regime have been designed in the past to implement somatic and synaptic functions [3]–[6]. However, with the increase of statistical variations due to the constantly decreasing feature size of transistors, performance of silicon neural networks requiring accurate setting of a "weight" parameter become strongly compromised. The typical solution for this problem is to increase transistor size. However, this alone might not be sufficient to guarantee good matching across the chip. For example, [7] demonstrates that $5\mu mX5\mu m$ transistor based 5-bit DACs fabricated in $0.35\mu m$ CMOS exhibit only 1.1 effective bits. Calibration techniques can be used to improve the accuracy; however, this incurs significant area penalty due to storage of calibration bits and is unacceptable in large scale systems. The problem of mismatch is also exacerbated for several nanometer scale non-CMOS devices (e.g. memristor [8]–[10] or domain wall magnets (DWM) [11]) that have potential for use in large scale neuromorphic applications. For example, DWM synapses are expected to have typical programming accuracies of 4-bits which can reduce to 2-3 bits effective resolution due to mismatch [11]. Hence, there is a strong need to develop algorithms and architectures that retain the performance of earlier spiking systems but require low-resolution weights.

In this paper, a hardware friendly morphological learning rule for neurons with nonlinear dendrites (NNLD) and binary synapses for classifying spatiotemporal spike patterns is presented. Previous studies have shown that NNLD can be successfully applied to learn both mean rate encoded inputs [12], [13] and spike-timing information [14], [15]. Although in [14], [15] we proposed a spike time based learning rule for training a NNLD architecture, however the rule was not optimized for memory capacity. Here we propose a novel memory capacity optimized spike timing based learning rule for training NNLD. Our work is inspired by the Tempotron learning rule for spiking neurons [1]. However, unlike the Tempotron learning rule that requires weights with high resolution, the proposed network uses low-resolution non-negative integer weights and learns through modifying connections of inputs to dendritic branches. Thus the 'morphology' or structure of the neuron (in terms of connectivity pattern) reflects the learning. This results in easier hardware implementation since a low-resolution non-negative integer weight of W can be implemented by activating a shared binary synapse W times through time multiplexing schemes like address event representation (AER) [16], [17]. Furthermore, the spiking threshold of the neuron employed in Tempotron is fixed throughout the learning. On the other hand, the proposed method is equipped with a threshold adaptation mechanism trying to optimize the number of false positives and false negatives. Some initial results of this concept were presented in [19]. In this paper, we present a novel threshold adaptation technique, more detailed set of results, comparisons with other work and application to a real world problem.

Financial Support from MOE through grant MOE2012-T2-2-054 is acknowledged.

Subhrajit Roy, Shaista Hussain and Arindam Basu are with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore (e-mail: subhrajit.roy@ntu.edu.sg; shaista001@e.ntu.edu.sg; arindam.basu@ntu.edu.sg). Phyo Phyo San is with the Institute for Infocomm Research, Singapore (e-mail: sanpp@i2r.a-star.edu.sg). Lee Wang Wei is with the SINAPSE laboratory at National University of Singapore (e-mail: lee_wang_wei@u.nus.edu). *Arindam Basu is the corresponding author.

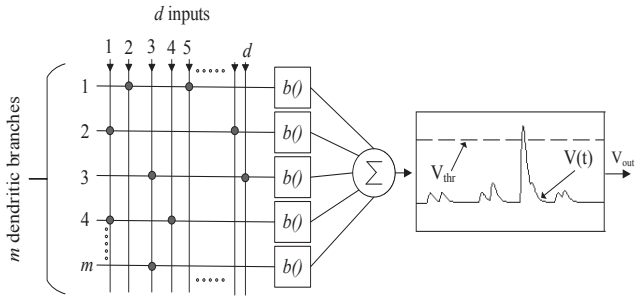

Fig. 1: The architecture of neuron with nonlinear dendrites (NNLD).

The organization of this paper is as follows: in Section II and III, the architecture of the neuron with nonlinear dendrites and its morphological learning algorithm are introduced. Two spike time based binary classification tasks are discussed in Section IVA and performance of our method in comparison to Tempotron is shown in Section IVB, IV-C and V. Finally, our work is compared with other classifiers in Section VI and then a conclusion is drawn in Section VII.

II. Neuron with Nonlinear Dendrites

It has been shown that a neuron with lumped dendritic nonlinearities possesses higher storage capacity than its counterparts with linear dendritic summation [12]. As presented in Fig. 1, the structure of NNLD is characterized by m identical dendritic branches and k excitatory synaptic contacts per branch. For each branch, the synaptic contact is formed by one of d dimensions of input afferents where $d >> k$. At the relevant times governed by incoming spikes, the synapses are activated and the membrane voltage is calculated by weighted sum of postsynaptic potentials (PSPs) as follows:

$$V(t) = \sum_{j=1}^{m} b(v_j(t)) = \sum_{j=1}^{m} b\left(\sum_{i=1}^{d} w_{ij} \sum_{f} K\left(t - t_f^i\right)\right) \quad (1)$$

where w_{ij} is the weight of the i^{th} synapse formed on the j^{th} branch, $v_j(t)$ is the input to the j^{th} dendritic nonlinearity, $b(\cdot)$ is the nonlinear activation function of the dendritic branch which is characterized by $b(v_j(t)) = {v_j(t)^2}/{x_{thr}}$ [13], K denotes the post-synaptic potential kernel and t_f^i are times of incoming spikes on the i^{th} afferent. Since we consider binary synapses, $w_{ij} \epsilon \{0, 1\}$. To suppress unrealistically large values, we include a saturation level x_{sat} at the output of each dendrite such that for $b(v_j(t)) > x_{sat}, b(v_j(t)) = x_{sat}$. Similar to our earlier work [13], we allow each input afferent to make multiple synaptic connections on the same dendritic branch but limit the total number of connections per branch by enforcing $\sum_{i=1}^{d} w_{ij} = k$ for each j. In this work, we consider normalized PSP of the form:

$$K(t - t_f) = V_0(\exp[-(t - t_f)/\tau] - \exp[-(t - t_f)/\tau_s]) \quad (2)$$

where the parameters τ and $\tau_s = \tau/4$ denote the decay time constants of membrane integration and synaptic current respectively [1].

For binary classification, the final output voltage V_{out} is interpreted to take a value of either "1" or "0" depending on whether the summed membrane voltage $V(t)$ is crossing a threshold voltage (V_{thr}). Similar to [1], this indicates that a neuron fires at least one spike if $V(t)$ crosses (V_{thr}); otherwise it remains quiescent. After the neuron fires a spike, it is returned to the refractory state and the rest of the spikes in the incoming pattern do not affect the computation.

III. Morphological Learning Algorithm

On one hand the proposed learning rule tries to optimize the connections between the input lines and the synapses while on the other hand it tries to find a optimal value of the neuron firing threshold(V_{thr}). Below we describe both the processes.

a) Learning the connections: Since we have binary synapses, a morphological learning rule that can modify the connections between afferent lines and synapses is needed. The inspiration of this work comes from the Tempotron learning rule [1] that learnt the temporal feature of random spike patterns in two classes by updating its weights so that the neuron fires a spike for patterns in class P^+ and stays silent for the other class. It was shown in [1] that the Tempotron rule endows a neuron with larger classification capacity than a perceptron with same number of synapses. Hence, we also start with a cost function that measures the deviation between the maximum membrane voltage (V_{max}) generated by misclassified patterns and V_{thr} defined as:

$$E = \begin{cases} V_{thr} - V_{max}, & if\ pattern\ in\ P^+\ is\ presented \\ V_{max} - V_{thr}, & if\ pattern\ in\ P^-\ is\ presented \end{cases} \quad (3)$$

where V_{max} is the maximal value of postsynaptic potential $V(t)$ at the time t_{max}, i.e., $V_{\max} = V(t_{\max})$. According to the gradient-descent method, the change in synaptic efficacy for the first case is calculated by:

$$\begin{aligned} \Delta w_{ij} &= -\frac{\partial E}{\partial w_{ij}} \\ &= -\frac{\partial\left(V_{thr} - \left(\sum_{j=1}^{m} b\left(\sum_{i=1}^{d} w_{ij} \sum_{f} K\left(t_{max} - t_f^i\right)\right)\right)\right)}{\partial w_{ij}} \\ &= b'(v_j(t_{max})) \sum_{f} K(t_{max} - t_f^i) \end{aligned} \quad (4)$$

where $b'(.)$ denotes the derivative of $b(.)$. The gradient for second case can be calculated similarly and we do not show it here for brevity. As mentioned earlier, since we consider binary weights, i.e., $w_{ij} = 1$ if a connection exists and 0 otherwise, we cannot directly modify the weights by adding the Δw_{ij} term derived here. Instead, the term Δw_{ij} in Equation 4 is reinterpreted as a correlation term $c_{ij} (= w_{ij})$ and is used to guide the process of swapping connections. At every iteration of the learning process, the synapse with the lowest c_{ij} averaged over an entire batch of patterns from a randomly chosen target set will be replaced (weight changed from 1 to 0) with the highest c_{ij} synapse in a candidate replacement set similar to [13]. To keep the paper self-contained, the mechanism of the learning process is outlined briefly below:

1) The learning process starts with random initialization of the connection matrix between input afferents and dendritic branches.

2) In each iteration of the training process, the activation of synapses on each dendritic branch are determined and the cell membrane output voltage ($V(t)$) in (1) is calculated for all the input patterns.
3) From the calculated $V(t)$, the maximum membrane voltage (V_{max}) is observed and classification result is determined, i.e., the patterns are correctly classified if $V_{\max} > V_{thr}$ for P^+ and $V_{\max} < V_{thr}$ for P^-.
4) A random set T of n_T synapses having weight 1 was targeted for possible replacement. For all misclassified patterns, the correlation term, c_{ij} is calculated for each synapse in T and averaged over the entire pattern set,
5) The poorest-performing synapse (minimum c_{ij}) in T, T_{min} is chosen for replacement.
6) To aid the replacement process, a randomly chosen set R containing n_R of the d afferent lines is forced to make silent synapses with weight 1 on the dendritic branch of T_{min}. These synapses are "silent" since they do not contribute PSP to the computation of $V(t)$–so they do not alter the classification when the same pattern set is re-applied. But now c_{ij} is calculated for synapses in R and T_{min} is replaced with the best-performing synapse (maximum c_{ij}) in R.
7) The learning from step (2) to (6) continues until all the patterns are correctly learnt (or) the maximum number of iteration is reached.

b) Learning the threshold: Since we do not have an arbitrary multiplicative weight in our neural model, the range of maximum voltages obtainable from our model in response to a fixed temporal spike pattern is limited. This is similar to the problem faced in [18]. Hence, improper selection of threshold may largely degrade the classification performance since a very large V_{thr} ($= m \times k \times K_{max}$ for example) may never be crossed by $V(t)$. In [19], we determined the value of V_{thr} by noting the maximum value of $V(t)$, i.e., V_{max} at time t_{max} for a large number of random input spike patterns and connection matrices. The resultant probability distribution of V_{max} was used to determine the optimal threshold V_{thr}. In [19], V_{thr} was set to be the voltage corresponding to the peak of probability distribution function. Here, we propose an automatic mechanism for adapting V_{thr} during training. This technique involves updating the value of V_{thr} after each iteration which is guided by the following formula:

$$\Delta V_{thr} = \eta(w_{fp}FP - w_{fn}FN) \tag{5}$$

where FP, FN, w_{fp}, w_{fn} and $\eta > 0$ are the number of false positives, number of false negatives, weightage associated with false positive error, weightage associated with false negative error and threshold learning rate respectively. In this article, we keep $w_{fp} = w_{fn} = 1$. Equation 5 is responsible for balancing the number of false positives and false negatives. When $FP > FN$, the number of negative patterns incorrectly classified as positive patterns is more than the number of positive patterns incorrectly classified as negative patterns so to balance FP and FN Equation 5 increases the value of V_{thr}. On the other hand, when $FP < FN$ Equation 5 diminishes the value of V_{thr}. The rate of threshold adaptation is controlled by the threshold learning rate η.

IV. Experiments and Results

A. Problem Description

In this sub-section, we describe the two tasks used to demonstrate the performance of our algorithm. The reason for this choice is that both of these are standard problems shown in [1] and facilitates comparison.

1) Task I: Classifying Random Latency Patterns: The first task is binary classification of single spike random latency patterns [1]. To perform the task, P spike patterns were generated and randomly assigned to one of the two classes P^+ (Class 1) or P^- (Class 2). Each spike pattern X $= (x_1, x_2, \ldots, x_d)$ consists of d afferents, where each of them spiked only once at a time drawn independently from a uniform distribution between 1 and T ms.

2) Task II: Classifying pairwise Synchrony Patterns: To examine the ability of our algorithm to learn correlations in multiple spikes, another data set that consists of pairwise synchrony events in each pattern is generated. In this data set, all the d afferents are grouped into $(d/2)$ pairs and afferents in a given pair fire single spike patterns synchronously. Since synchronous events occur at random, uniformly distributed times in both pattern categories, so that class information is embedded solely in the patterns of synchrony; neither spike counts nor spike timing of individual neurons carry any information relevant for the classification task. This task mimicked spike synchrony-based sensory processing.

For both Task I and II, we have kept d as 500.

B. Results: Performance of NNLD trained by morphological learning algorithm

Throughout the experiment, the design parameters m, k, x_{thr}, x_{sat}, τ and T, were chosen as 100, 5, 1, 100, 15 ms and 400 ms respectively. The detailed criteria for selection of parameters x_{thr} and x_{sat} is described in [13]. As discussed in Section III, the firing threshold V_{thr} is adapted and so it is randomly initialized before training.

To start with Task I, the NNLD in Fig. 1 was trained on a small number ($= 100$) random latency patterns as generated in Section IV-A1. The results in Fig. 2 (a) and (b) show that the proposed method can efficiently perform the classification task. A clear separation between Class 1 and Class 2 shows that the proposed method is able to respond to the random single spike latency patterns by shifting V_{max} closer and farther to the V_{thr} for each pattern in classes 1 and 2 respectively.

The NNLD was also trained for larger number of input patterns (500 and 1000 patterns) as presented in Fig. 3 (b) and (c). It shows that the proposed method can perform the classification task quite well by achieving accuracies of $95.58\%(SD = 0.54\%)$ and $86.57\%(SD = 0.72\%)$ respectively for these cases. Next, we performed the experiment of Task II with our network. It was observed that the classification performances (100%, 100% and 99.71 $\%(SD = 0.1\%)$ accuracy for 100, 500 and 1000 patterns) are much better than that of performance in random latency patterns of Task I.Hence, our method can identify the extra information embedded in synchrony of neural firings. It also reveals that the learning

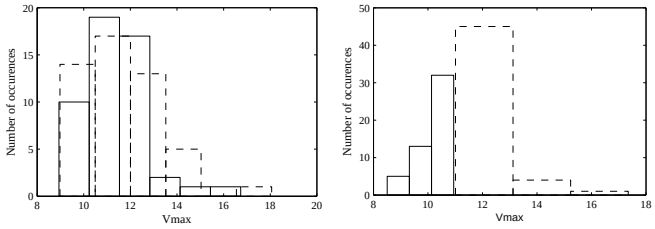

Fig. 2: Distribution of V_{max} for 100 patterns (a)before and (b) after training for Task I. V_{max} for patterns in P^+ and P^- are shown in dashed and solid line respectively.

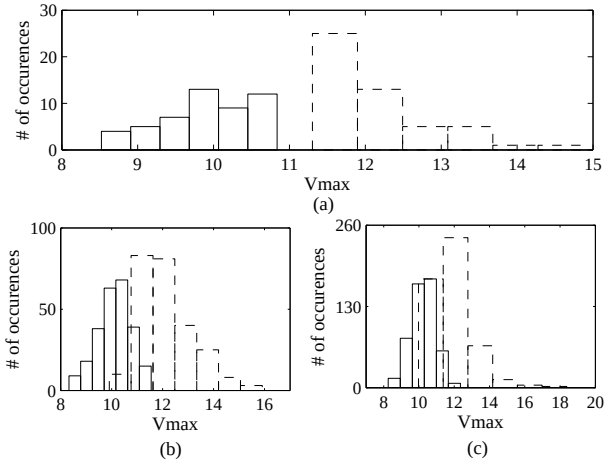

Fig. 3: The distribution of V_{max} for a set of (a) 100, (b) 500 and (c) 1000 patterns in task I showing 100%, 94.8% and 84.9% accuracy respectively.

rule does not depend only on a single synchronous pair but on a huge number of synchronous pairs.

C. Results: Comparison with Tempotron learning

Next, the performance of proposed method is compared with the Tempotron learning [1] that learns spike time patterns by using weight updates. The number of synapses used by Tempotron is equal to the number of input afferents d. Since we are interested in the performance of these algorithms in their hardware implementations plagued by mismatch, we consider the performance of the Tempotron when its weight is quantized at different resolutions. Further, we do the quantization in two ways: either after training (AT) or as a step during training (DT). The first method corresponds to the case where weights trained in software version of the algorithm is downloaded to the hardware while the second method is analogous to performing training on-chip. Furthermore, for comparison with our previous threshold selection technique [19], we have also calculated the value of V_{thr} by equating it to the voltage corresponding to the peak location of V_{max} distribution over 10000 random input spike patterns. We term this as $V_{thr,static}$. Fig. 4 depicts that the performance obtained by morphological learning with adaptive threshold is superior to that of learning with fixed $V_{thr,static}$. Moreover, the comparison results in Fig. 4 (a) also show that the Tempotron using floating-point numbers achieves better performance compared to the

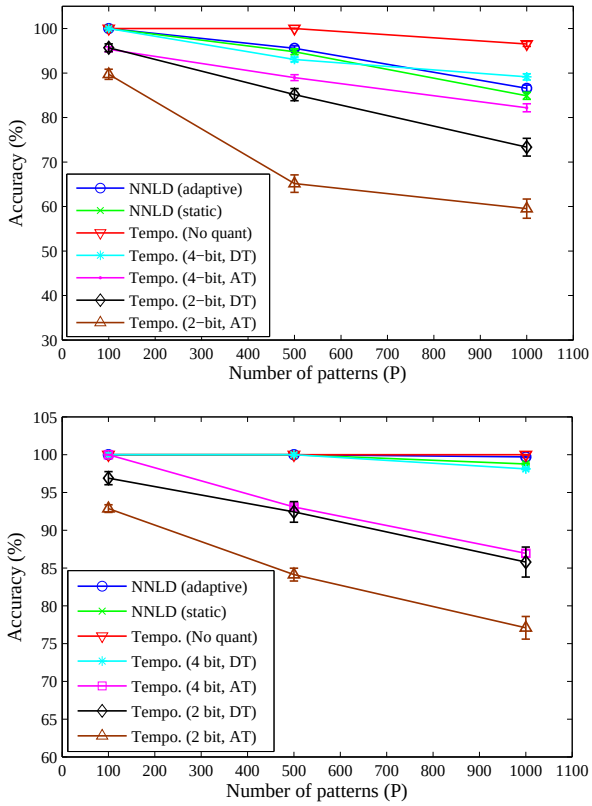

Fig. 4: Comparison studies on classification performance, for Tempotron learning [1] and the proposed morphological learning rule for (a) Task I of Random Latency Patterns and (b) Task II of Pair-Wise Synchrony Patterns. The results have been averaged over 10 independent trials.

proposed method. However, when the high resolution weights are quantized at 2-bit level, its performance is worse than the proposed method. Also, it can be seen that the performance is better when the quantization is performed within the learning loop. This is to be expected since the learning algorithms can now try to correct this quantization error as well.

Only at 4-bit quantization level, the classification performance of Tempotron ($93.05\%(SD = 0.55\%)$ and $89.14\%(SD = 0.7\%)$ accuracy for 500 and 1000 patterns) in Task I is comparable to our proposed method using 1-bit or binary weights. This underlines the importance of our proposed method in robustly implementing spike timing based classifiers using low-resolution analog synapses available in nano-scale CMOS or non-CMOS devices. Similarly, the classification performance of NNLD for Task II with 1-bit binary weights in Fig. 4 (b) is comparable to performance of Tempotron at 4-bit quantization which shows about 100 % and $98.12\%(SD = 0.12\%)$ accuracy respectively for 500 and 1000 patterns.

V. Application Example

Till now the proposed algorithm has been applied to classify synthetic synchrony patterns. But, to check whether the algorithm can work in real world problems, it is used to classify Tactile information. The patterns which have been till now presented for classification had one spike per afferent but real world scenarios may have multiple spikes per afferent.

TABLE I: Performance comparison of morphological learning rule on NNLD and Tempotron

Cases	Accuracy	
	Mean	SD
Morph. lear. on NNLD (adaptive V_{thr})	96.54	0.6573
Morph. lear. on NNLD (for V_{thr_static})	95.64	0.6473
Tempotron (no quantization) [1]	97.06	0.5123
Tempotron (6-Bit quantization after training)	95.64	0.7614
Tempotron (6-Bit quantization during training)	96.26	0.6822
Tempotron (4-Bit quantization after training)	78.87	1.8371
Tempotron (4-Bit quantization during training)	85.92	1.3846
Tempotron (2-Bit quantization after training)	51.56	3.765
Tempotron (2-Bit quantization during training)	55.88	2.8731

a) Task Description: A detailed description of the task can be found in [20]–here, we give a brief description for completeness. The task requires a flexible, stretchable and conformable tactile sensor array made up of conductive fabric which is used for data collection. Two glass spheres (indenters) of diameter 65 mm and 105 mm were indented onto this sensor array by a suitably programmed 6-axis robotic arm. The indentation force used was 4N, as measured using a load cell placed below the sensor having an accuracy of 0.01N. The signal recording was started just before placing the indenter onto the sensor array and stopped before removing the indenter. Between consecutive indentations, a 5 second pause was provided so that the sensors were able to recover. A total of 100 recordings were taken per indenter. The collected data were converted to spikes as described in the next sub-section. For each indenter, 60 randomly chosen recorded data was used for training both the proposed algorithm and Tempotron. The remaining 40 recordings were used for testing.

b) Spike Train Generation: The conversion of the analog data recorded by the sensor array to spike trains involves the following steps. First, the analog data is converted to digital output by applying a fixed threshold of 0.5. This digital output is inverted to generate another set of digital data. Each channel of the digital data and its inverted version are passed through Leaky Integrate and Fire Neuron to generate two sets of spike trains. These spike trains are combined to form the spike response to be given as an input to the algorithms. The usage of both the digital data and its inverted version for spike generation ensures both low to high and high to low transitions are captured by a change in firing activity. The combined spike train patterns given as input to the algorithms consist of 130 afferents. Thus, we allot 130 synapses for both Tempotron and NNLD.

c) Results: The performance of the proposed method on this application is compared with Tempotron algorithm at different quantization levels in Table I. The results, averaged over 10 independent trials, show that the proposed algorithm is able to achieve an accuracy of $96.54\%(SD = 0.6543\%)$. Although Tempotron without quantization performs better ($97.06\%(SD = 0.5123\%)$) than the proposed learning rule on NNLD, but after quantization, at least 6 bits of weight resolution is needed by Tempotron to match our performance with 1 bit weights.

VI. DISCUSSION

Here, we compare our method with other reported algorithms and other possible variants of morphological learning.

A. Comparison to other supervised spiking neural classifiers

In recent years, several supervised learning algorithms have been proposed such as Tempotron [1], ReSuMe [21], SpikeProp [22] and Chronotron [23] for training spiking neurons. From a pattern recognition viewpoint, these algorithms can be classified into two types. The general theme of the first type of algorithms is that a desired output spike train is specified prior to learning for each class of patterns. SpikeProp, ReSuMe and Chronotron are examples of this type, among which SpikeProp can only produce a single output spike whereas the others are capable of producing multi-spike train. In the second type, no such desired output spike train is specified beforehand and the algorithms choose the best time to spike for each pattern during training. Tempotron and the proposed algorithm are examples of this type which chooses t_{max} (defined in Sec. III) as the time to spike. We have already compared the performance of our algorithm with Tempotron. We now choose ReSuMe as a representative of the first type of algorithms and compare its performance with the proposed learning rule for the two tasks described in Sec. IV-A. For ReSuMe, the neuron had to fire one spike at $t_+ = 350ms$ for P^+ patterns and at $t_- = 450ms$ for P^- patterns. The classification performance of ReSuMe (89.58%(SD=1.24%)) and 82.65%(SD=2.22%) accuracy for 500 and 1000 patterns) in Task I is worse compared to the proposed method. Similarly, for Task II, the proposed method performs much better than ReSuMe learning rule ($96.74\%(SD = 0.8467\%)$ and $92.12\%(SD = 1.1654\%)$ accuracy for 500 and 1000 patterns).

B. Comparison to other classifiers using dendritic processing

We have already compared the proposed algorithm with previous works employing NNLD in Section I. Here we first compare our method with another dendritic algorithm proposed by Wu et al. in [24]. Unlike [24] which considered only mean rate encoded inputs, our learning rule can be applied to arbitrary spike trains.

Second, we compare the proposed algorithm with another recently reported structural plasticity based algorithm named Dendritically Enhanced Readout (DER) [14], [15]. In [14], [15], this structure has been used as the readout stage of Liquid State Machine. The primary difference of the proposed algorithm with DER is in the number of data points per pattern to be memorized for a particular task. If we consider there are P^+ and P^- patterns of Class 1 and Class 2 respectively, then the number of data points to be memorized by the proposed algorithm are $P_{m\_NNLD} = T \times P^- + 1$ while the number of data points to be memorized by DER are $P_{m\_DER} = T \times (P^+ + P^-)$ where T is the number of time points per pattern. So, DER has to memorize almost two times more data points than the proposed algorithm for the same number of patterns thereby reducing its memory capacity.

Third, we have also compared our performance with another recently proposed dendritic algorithm termed as Synaptic Kernel Inverse Method(SKIM) [25]. The difference mentioned above in the context of DER also applies to SKIM. Apart from that, SKIM also uses much more resources than the proposed NNLD which we will show next. We compare the two methods

on Task I for 100 patterns in which case our proposed method has 100% accuracy. The neural network architecture used in [25] consists of N presynaptic neurons which connect to an output spiking neuron, via synaptic connections to its M dendritic branches. The weights of these synapses are random and fixed. These synapses along with a subsequent nonlinearity projects the input to a higher dimension thereby increasing separability. The dendritic branches sum the synaptic input currents, and the output from the dendritic branches are summed at the soma of the output neuron. The weights of the connection between dendritic branches and the soma are learnt by Moore pseudo inversion method [25]. Thus, for this network $N \times M$ synaptic resources are required for connecting the N presynaptic neurons to M postsynaptic dendritic branches and M synapses are required for connecting the M dendritic branches to the single output spiking neuron. The number of input spiking neurons are equal to the number of afferents in the input data which in our case is 500. Since we have used 500 synapses for NNLD, so initially we keep the number of postsynaptic dendrites in SKIM as 1 ($M = 1$) to match the number of resources used by us. But for $M = 1$, SKIM fails miserably and provides only 50% accuracy. When the number of dendritic branches are increased, SKIM provides better results and finally is able to provide 100% accuracy when $M = 360$. So to provide equivalent result as the proposed algorithm SKIM requires 361 times more resources than our proposed NNLD.

VII. CONCLUSION

A morphological learning rule that can be used to find the optimal morphology of neurons with nonlinear dendrites (NNLD) and binary synapses is presented. The learning rule includes a novel threshold adaptation technique. To see the effectiveness of the proposed method, the NNLD trained with morphological rule is used to solve two classification tasks and one real world problem. The results depict that our proposed method with 1 bit weights can achieve comparable performance to tempotron learning rule with 4-bit to 6-bit quantized weights.

ACKNOWLEDGEMENT

The authors acknowledge helpful discussions with Prof. Nitish Thakor regarding tactile sensing.

REFERENCES

[1] R. Gutig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing-based decisions", *Nature Neuroscience*, vol. 9, no. 3, pp. 420-428, 2006.

[2] R. Johansson and I. Birznieks, "First spikes in ensembles of human tactile afferents code complex spatial fingertip events", *Nature Neuroscience*, vol. 7, no. 3, pp. 170-177, 2004.

[3] A. Basu and C. Petre, and P. Hasler, "Bifurcations in a Silicon Neuron", *Proceedings of the International Symposium on Circuits and Systems*, pp. 428-431, 2013.

[4] J. Arthur and K. Boahen, "Synchrony in silicon: The gamma rhythm", *IEEE Transactions on Neural Networks*, no. 6, pp. 1815-1825, Nov. 2007.

[5] S. Millner, A. Grübl, K. Meier, J. Schemmel and M. O. Schwartz, "A VLSI Implementation of the Adaptive Exponential Integrate-and-Fire Neuron Model," *Advances in Neural Information Processing Systems*, Vancouver, B.C., Canada, Dec. 2010, pp. 1642-1650.

[6] F. Grassia, L. Buhry, T. Lévi, J. Tomas, A. Destexhe, and S. Saïghi, "Tunable Neuromimetic Integrated System for Emulating Cortical Neuron Models," *Frontiers in Neuroscience*, vol. 5, no. 134, 2011. doi:10.3389/fnins.2011.00134

[7] B. Linares-Barranco, T. Serrano-Gotarredona and R. Serrano-Gotarredona, "Compact Low-power Calibration Mini-DACs for Neural Arrays with Programmable Weights," *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1207-15, Nov. 2003.

[8] D. Querlioz, O. Bichler, P. Dollfus and C. Gamrat,"Immunity to Device Variations in a Spiking Neural Network with Memristive Nanodevices," *IEEE Transactions on Nanotechnology*, vol. 12, no. 3, pp. 288-95, May 2013.

[9] K. H. Kim, S. Gaba, D. Wheeler et.al., "A functional hybrid memristor crossbar-Array/CMOS system for data storage and neuromorphic applications," *Nano Letters*, vol. 12, no. 1, pp. 389-395, 2011.

[10] G. Indiveri, B. linares-Barranco, R. Legenstein, G. Deligeorgis and T. Prodromakis, "Integration of nanoscale memristor synapses in neuromorphic computing architectures," *Nanotechnology*, vol. 24, no. 38, Sep. 2013.

[11] M. Sharad, C. Augustine, G. Panogopoulos and K. Roy, "Spin-Based Neuron model with Domain-Wall Magnets as Synapse," *IEEE Transactions on Nanotechnology*, vol. 11, pp. 843-53, Jul. 2012.

[12] P. Poirazi and B. W. Mel, "Impact of active dendrites and structural plasticity on the memory capacity of neural tissue", *Neuron*, vol. 9, no. 3, pp. 779-776, 2001.

[13] S. Hussain, R. Gopalakrishnan, A. Basu and S. C. Liu, "A Morphological Learning: Increased Memory Capacity of Neuromorphic Systems with Binary Synapses exploiting AER based Reconfiguration", *Proceedings of the International Joint Conference on Neural Networks*, Dallas, Texas, Aug. 2013, pp. 1-7.

[14] S. Roy, A. Basu and S. Hussain, "Hardware efficient, neuromorphic dendritically enhanced readout for liquid state machines", *Proceedings of the Biomedical Circuits and Systems Conference*, Rotterdam, The Netherlands, Oct. 2013, pp. 302-305.

[15] S. Roy, A. Banerjee and A. Basu, "Liquid State Machine With Dendritically Enhanced Readout for Low-Power, Neuromorphic VLSI Implementations," *IEEE Transactions on Biomedical Circuits and Systems*, vol.8, pp.681-695, Oct. 2014.

[16] K. Boahen, "Point-to-point connectivity between neuromorphic chips using address events", *IEEE Transactions on Circuits and Systems II*, vol. 47, pp. 416-434, May 2000.

[17] S. Brink, S. Nease, P. Hasler, S. Ramakrishnan, R. Wunderlich, A.Basu and B. Degnan, "A learning-enabled neuron array IC based upon transistor channel models of biological phenomenon", *IEEE Transactions on Biomedical Circuits and Systems*, vol. 7, Mar. 2013, pp. 71-81.

[18] S. Hussain, A. Basu, M. Wang and T. Hamilton, "DELTRON: Neuromorphic architectures for delay based learning", *Proceedings of the Asia Pacific Conference on Circuits and Systems*, Kaohsiung, Taiwan, Dec. 2012, pp. 304-307.

[19] P. P. San, A. Basu and S. Hussain, "Spike-timing dependent morphological learning for a neuron with nonlinear active dendrites", in *Proceedings of the IEEE IJCNN*, Beijing, China, Jul. 2014, pp. 3192 - 3196.

[20] W. W. Lee, J. Cabibihan and N. V. Thakor , "Bio-mimetic strategies for tactile sensing", in *Proceedings of IEEE Sensors Conference*, Baltimore, MD, Nov. 2013, pp. 1-4.

[21] F. Ponulak and A. J. Kasinski, "Supervised Learning in Spiking Neural Networks with ReSuMe: Sequence Learning, Classification, and Spike Shifting.', *Neural Computation*, vol. 22, no. 2, pp. 467-510, 2010.

[22] S. M. Bohte, J. N. Kok, and H. La Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons", *Neurocomputing*, vol.48, no. 1, pp. 17-37, 2002.

[23] R. V. Florian, "The chronotron: a neuron that learns to fire temporally precise spike patterns", *PLoS ONE*, vol. 7, no. 8, e40233, 2013.

[24] X.E. Wu and B.W. Mel, "Capacity-enhancing learning rules in a medial temporal lobe online learning model", *Neuron*, vol. 62, no. 1, pp. 31-41, 2009.

[25] J. C. Tapson, G. K. Cohen, S. Afshar et al., "Synthesis of neural networks for spatio-temporal spike pattern recognition and processing", *Frontiers in Neuroscience*, vol. 7, no. 153, Aug. 2013. doi: 10.3389/fnins.2013.00153